# 交錯區塊DES
# The Cross-Block DES

葉義雄 劉雪櫻 謝志敏 陳維魁
Yi-Shiung Yeh Hsueh-Ying Liu Tsu-Miin Hsieh Wei-Kuei Chen

國立交通大學資訊工程系所
Department of Computer Science and Information Engineering
National Chiao Tung University, Hsin Chu, Taiwan, R.O.C

**摘 要**

*在本篇論文中，我們針對DES的加密區塊大小提出兩種變化，其加密區塊分別為128及32*q (q≥2)位元。由於加密的區塊大小增加，使得其安全性大為提高。另外，在我們設計這個加密方法時，發現有些區塊的交錯方式不存在解密方法。*

**關鍵字：** 加密，解密，美國資料加密標準 (DES)

**Abstract**

*In this paper, we make some changes to the block size of DES. The original DES encrypts 64 bits of plaintext. Here we propose two variants. The encryption block sizes are 128 and $32 \times q (q \geq 2)$ bits, respectively. The algorithms are more secure as the size of encryption block increases. Besides, we also find out that there does not exist decryption algorithms for some variants of the cross-block.*

**Keywords:** encryption, decryption, Data encryption standard (DES)

## 1. Introduction

DES was proposed by IBM and adopted as a federal standard on November 23, 1976 [1]. The original DES operates on 64-bit blocks of plaintext. Our variants try to expand the size of encryption block to be variable. There are some similar variants before. In Generalized DES (GDES)[1,2], it operates on variable-sized blocks of plaintext. The plaintext is divided into $q$ 32-bit sub-blocks and the number of rounds is variable. In each round and the function $f$ is calculated once on the right-most block. The result is XORed with all the other sub-blocks, i.e., $B_0^{(1)}, B_0^{(2)}, \ldots,$ and $B_0^{(q)}$, which are then rotated to the right one sub-block. The operations in the last round are modified slightly so that the encryption and decryption process differs only in the order of the subkeys. In figure 1.1, the operations of the round $i$ are shown.

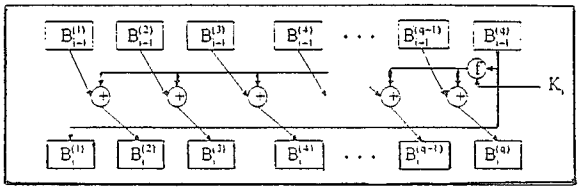

Figure 1.1 GDES

In the following sections we shall proposed two variants.

## 2. Variant I

### 2.1 Encryption

The first proposed variant, we call it variant I which operates on a 128-bit block of plaintext with two 56-bit keys. It makes no differences between encryption and decryption algorithm except the order of the subkeys. While encryption, the 128-bit input block is first transposed under an initial permutation which is presented in Table 2.1. The generation of our initial permutation table is similar to the one in DES except the size expended from 56 to 128 and the difference between neighboring numbers is 16 instead of 8. The table should be read from left to right, top to bottom. For example, the initial permutation moves bit 114 of the plaintext to bit position 1, bit 118 to bit position 17, bit 122 to bit position 34, and so forth. Then the output of initial permutation is broken into 4 sub-blocks, each 32 bits long, which are passed through $n$ (n must be odd) rounds of transformations. After that, the 4 sub-blocks are joined together and transposed under the final permutation shown in Table 2.2. The final permutation must be the inverse of the initial permutation.

| | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 114 | 98 | 82 | 66 | 50 | 34 | 18 | 2 | 116 | 100 | 84 | 68 | 52 | 36 | 20 | 4 |
| 118 | 102 | 86 | 70 | 54 | 38 | 22 | 6 | 120 | 104 | 88 | 72 | 56 | 40 | 24 | 8 |
| 122 | 106 | 90 | 74 | 58 | 42 | 26 | 10 | 124 | 108 | 92 | 76 | 60 | 44 | 28 | 12 |
| 126 | 110 | 94 | 78 | 62 | 46 | 30 | 14 | 128 | 112 | 96 | 80 | 64 | 48 | 32 | 16 |
| 113 | 97 | 81 | 65 | 49 | 33 | 17 | 1 | 115 | 99 | 83 | 67 | 51 | 35 | 19 | 3 |
| 117 | 101 | 85 | 69 | 53 | 37 | 21 | 5 | 119 | 103 | 87 | 71 | 55 | 39 | 23 | 7 |
| 121 | 105 | 89 | 73 | 57 | 41 | 25 | 9 | 123 | 107 | 91 | 75 | 59 | 43 | 27 | 11 |
| 125 | 109 | 93 | 77 | 61 | 45 | 29 | 13 | 127 | 111 | 95 | 79 | 63 | 47 | 31 | 15 |

Table 2.1 Initial permutation

| | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 72 | 8 | 80 | 16 | 88 | 24 | 96 | 32 | 104 | 40 | 112 | 48 | 120 | 56 | 128 | 64 |
| 71 | 7 | 79 | 15 | 87 | 23 | 95 | 31 | 103 | 39 | 111 | 47 | 119 | 55 | 127 | 63 |
| 70 | 6 | 78 | 14 | 86 | 22 | 94 | 30 | 102 | 38 | 110 | 46 | 118 | 54 | 126 | 62 |
| 69 | 5 | 77 | 13 | 85 | 21 | 93 | 29 | 101 | 37 | 109 | 45 | 117 | 53 | 125 | 61 |
| 68 | 4 | 76 | 12 | 84 | 20 | 92 | 28 | 100 | 36 | 108 | 44 | 116 | 52 | 124 | 60 |
| 67 | 3 | 75 | 11 | 83 | 19 | 91 | 27 | 99 | 35 | 107 | 43 | 115 | 51 | 123 | 59 |
| 66 | 2 | 74 | 10 | 82 | 18 | 90 | 26 | 98 | 34 | 106 | 42 | 114 | 50 | 122 | 58 |
| 65 | 1 | 73 | 9 | 81 | 17 | 89 | 25 | 97 | 33 | 105 | 41 | 113 | 49 | 121 | 57 |

Table 2.2 Final permutation

In this variant, the operations are different in the even and odd rounds. Let $B_{i-1}^{(j)}$ denote the $j$th input sub-block of the $(i-1)$th round, $B_i^{(j)}$ denote the $j$th output sub-block of the $i$th round. In each odd round (see Figure 2.1), the sub-block $B_{i-1}^{(2)}$ is operated under function $f$ with the subkey $K_i^{(1)}$ and then XORed with the sub-block $B_{i-1}^{(1)}$. The result becomes the sub-block $B_i^{(2)}$. The sub-block $B_{i-1}^{(2)}$ becomes the sub-block $B_i^{(1)}$. The generations of $B_i^{(3)}$ and $B_i^{(4)}$ are similar to the first two sub-blocks. We can see it in this figure.

In each even round (see Figure 2.2), the sub-block $B_{i-1}^{(4)}$ becomes the sub-block $B_i^{(1)}$. The sub-block $B_{i-1}^{(3)}$ becomes the sub-block $B_i^{(2)}$. The sub-block $B_{i-1}^{(3)}$ is operated under function $f$ with the subkey $K_i^{(1)}$ and then XORed with the sub-block $B_{i-1}^{(2)}$. The result becomes the sub-block $B_i^{(3)}$. The sub-block $B_{i-1}^{(4)}$ is operated under function $f$ with the subkey $K_i^{(2)}$ and then XORed with the sub-block $B_{i-1}^{(1)}$. The result becomes the sub-block $B_i^{(4)}$.

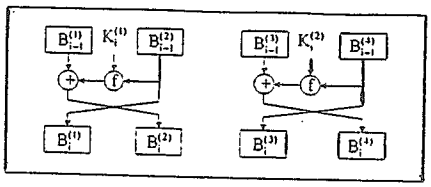

Figure 2.1 The operations of odd round in variant I

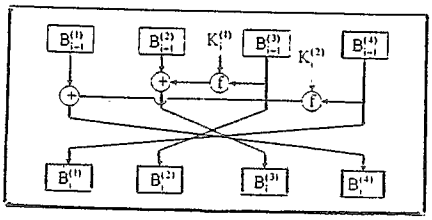

Figure 2.2 The operations of even round in variant I

To make the decryption algorithm be the same as the encryption algorithm, the subblocks of the final round must be reordered by $B_n^{(4)}B_n^{(3)}B_n^{(2)}B_n^{(1)}$. The encryption algorithm is described as follows.

***Algorithm 2.1*** Encryption of variant I
Input: Plaintext $P$, subkeys
$K_1^{(1)}, K_2^{(1)}, \ldots, K_n^{(1)}$ and $K_1^{(2)}, K_2^{(2)}, \ldots, K_n^{(2)}$
Output: Ciphertext $C$
Process:
$P$ is transposed under $IP$
$IP(P)$ is divided into $B_0^{(1)}$, $B_0^{(2)}$, $B_0^{(3)}$, $B_0^{(4)}$
For $i$=1 to $n$
Begin
If $i$ is even
Begin

$$B_i^{(1)} = B_{i-1}^{(2)}$$
$$B_i^{(2)} = B_{i-1}^{(1)} \oplus f(B_{i-1}^{(2)}, K_i^{(1)})$$
$$B_i^{(3)} = B_{i-1}^{(4)}$$
$$B_i^{(4)} = B_{i-1}^{(3)} \oplus f(B_{i-1}^{(4)}, K_i^{(2)})$$

End
else
Begin

$$B_i^{(1)} = B_{i-1}^{(4)}$$
$$B_i^{(2)} = B_{i-1}^{(3)}$$
$$B_i^{(3)} = B_{i-1}^{(2)} \oplus f(B_{i-1}^{(3)}, K_i^{(1)})$$
$$B_i^{(4)} = B_{i-1}^{(1)} \oplus f(B_{i-1}^{(1)}, K_i^{(2)})$$

End
End
$C = IP^{-1}(B_n^{(4)}, B_n^{(3)}, B_n^{(2)}, B_n^{(1)})$

□

The overall structure is shown in Figure 2.3. The number of rounds must be odd such that the encryption and decryption algorithm are the same except the order of subkeys used in decryption and encryption is different. The two initial keys $K^{(1)}$ and $K^{(2)}$ are used to generate subkeys $K_i^{(1)}$ and $K_i^{(2)}$ $(1 \le i \le n)$ respectively. The generation of subkeys is similar to the one in DES except that the number of generated subkeys is different. The key scheduling of DES generates 16 subkeys because there are 16 rounds in DES. As the number of rounds $n$ in variant I is not 16, we must design a different left shift bits table. This can be done by doing some changes to the left shift bits table of DES. If $n$ is greater than 16, we can increase the size of the table by splitting any round that left shift 2 bits to two rounds that left shift 1 bit; if $n$ is less than 16, we can decrease the size of the table by combining the left shift bits two rounds into one round. Here we generate a left shift table which produces 15 subkeys in table 2.3 by combining the first two columns of the original table. Variant I may use only one 56-bit key which

means $K^{(1)} = K^{(2)}$. It's obviously that variant I with one 56-bit key would be less secure than the one with two 56-bit keys.

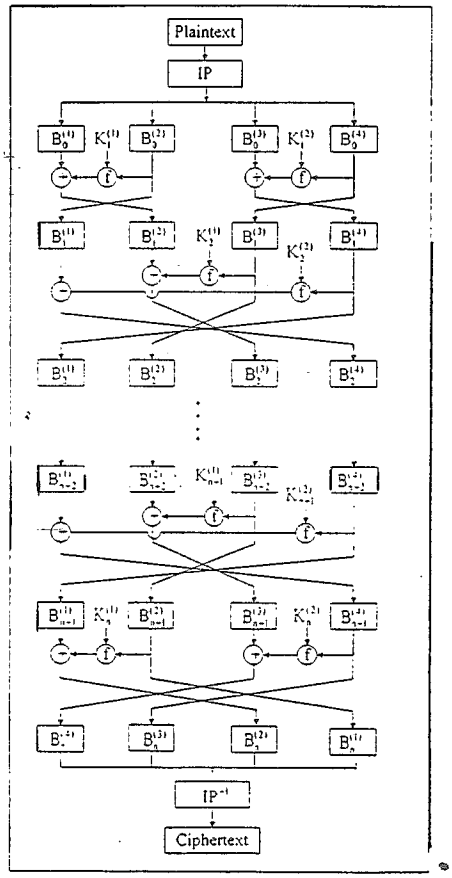


Figure 2.3 The overall structure of variant I

| Round | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Number | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 1 | 2 | 2 | 2 | 2 | 2 | 2 | 1 |

TABLE 2.3 Left shift number table

## 2.2 Decryption

The decryption algorithm is the same as the encryption algorithm except the order of the subkeys. The order of the encryption subkeys are $K_1^{(1)}, K_2^{(1)}, \ldots, K_n^{(1)}$ and; $K_1^{(2)}, K_2^{(2)}, \ldots, K_n^{(2)}$ the order of the decryption subkeys are changed and reversed, i.e., $K_n^{(2)}, K_{n-1}^{(2)}, \ldots, K_1^{(2)}$ and $K_n^{(1)}, K_{n-1}^{(1)}, \ldots, K_1^{(1)}$. Let $DK_j^{(i)}$ denote the $j$th key of the $i$th round in the decryption and $K_j^{(i)}$ is the $j$th key of the $i$th round in the encryption, then the relation between the encryption subkeys and decryption subkeys can be expressed as:

$$DK_j^{(i)} = K_{n+1-i}^{(3-j)} \ (j = 1, 2, \ 1 \le i \le n) \quad (1)$$

***Theorem 2.1*** Let $DB_i^{(j)}$ denote the $j$th sub-block of the $i$th round in the decryption and $DB_1^{(1)}, DB_0^{(2)}, DB_0^{(3)}, DB_0^{(4)}$ are the initial sub-block of ciphertext. We have

$$DB_0^{(j)} = B_n^{(5-j)}, \text{ for } 1 \le j \le 4 \quad (2)$$

In the following context, we prove the equation

$$DB_i^{(j)} = B_{n-i}^{(5-j)}, \text{ for } 1 \le i \le n \text{ and } 1 \le j \le 4 \ (3)$$

***proof:***

We will prove it by using mathematical induction

Basic induction: $i$=1, we will prove $DB_1^{(j)} = B_{n-1}^{(5-j)}$, for $1 \le j \le 4$.

According to the encryption/decryption algorithm and Eq.(1),(2), we have:

$$DB_1^{(1)} = DB_0^{(2)} = B_n^{(3)} = B_{n-1}^{(4)} = B_{n-1}^{(5-1)}$$

$$\begin{aligned} DB_1^{(2)} &= DB_0^{(1)} \oplus f(DB_0^{(2)}, DK_1^{(1)}) \\ &= B_n^{(4)} \oplus f(B_n^{(3)}, DK_1^{(1)}) \\ &= B_{n-1}^{(3)} \oplus f(B_{n-1}^{(4)}, K_n^{(2)}) \oplus f(B_{n-1}^{(4)}, DK_1^{(1)}) \\ &= B_{n-1}^{(3)} = B_{n-1}^{(5-2)} \end{aligned}$$

$$DB_1^{(3)} = DB_0^{(4)} = B_n^{(1)} = B_{n-1}^{(2)} = B_{n-1}^{(5-3)}$$

$$\begin{aligned} DB_1^{(4)} &= DB_0^{(3)} \oplus f(DB_0^{(4)}, DK_1^{(2)}) \\ &= B_n^{(2)} \oplus f(B_n^{(1)}, DK_1^{(1)}) \\ &= B_{n-1}^{(1)} \oplus f(B_{n-1}^{(2)}, K_n^{(1)}) \oplus f(B_{n-1}^{(2)}, DK_1^{(2)}) \\ &= B_{n-1}^{(1)} = B_{n-1}^{(5-4)} \end{aligned}$$

Hypothesis: let $i = k$ and the result holds. That is, $DB_k^{(j)} = B_{n-k}^{(5-j)}$, for $1 \le j \le 4$

Consider $i = k+1$,

if $k$ is even,

$$DB_k^{(1)} = DB_k^{(2)} = B_{n-k}^{(3)} = B_{n-k-1}^{(4)} = B_{n-(k+1)}^{(5-1)}$$

$$\begin{aligned} DB_k^{(2)} &= DB_k^{(1)} + f(DB_k^{(2)}, DK_{k+1}^{(1)}) \\ &= B_{n-k}^{(1)} \oplus f(B_{n-k}^{(3)}, DK_{k+1}^{(1)}) \\ &= B_{n-k-1}^{(3)} \oplus f(B_{n-k-1}^{(4)}, K_k^{(2)}) \oplus f(B_{n-k-1}^{(4)}, DK_{k+1}^{(1)}) \\ &= B_{n-k-1}^{(3)} = B_{n-(k+1)}^{(5-2)} \end{aligned}$$

$$DB_{k+1}^{(3)} = DB_k^{(4)} = B_{n-k}^{(1)} = B_{n-k-1}^{(2)} = B_{n-(k+1)}^{(5-3)}$$

$$\begin{aligned} DB_{k+1}^{(4)} &= DB_k^{(3)} \oplus f(DB_k^{(4)}, DK_{k+1}^{(2)}) \\ &= B_{n-k}^{(2)} \oplus f(B_{n-k}^{(1)}, DK_{k+1}^{(2)}) \\ &= B_{n-k-1}^{(1)} \oplus f(B_{n-k-1}^{(2)}, K_n^{(1)}) \oplus f(B_{n-k-1}^{(2)}, DK_{k+1}^{(2)}) \\ &= B_{n-k-1}^{(1)} = B_{n-(k+1)}^{(5-4)} \end{aligned}$$

else

$$DB_{k+1}^{(1)} = DB_k^{(4)} = B_{n-k}^{(1)} = B_{n-k-1}^{(4)} = B_{n-(k+1)}^{(5-1)}$$

$$\begin{aligned} DB_{k+1}^{(4)} &= DB_k^{(1)} \oplus f(DB_k^{(1)}, DK_{k+1}^{(2)}) \\ &= B_{n-k}^{(4)} \oplus f(B_{n-k}^{(1)}, DK_{k+1}^{(2)}) \\ &= B_{n-k-1}^{(1)} \oplus f(B_{n-k-1}^{(4)}, K_{n-k}^{(1)}) \oplus f(B_{n-k-1}^{(4)}, DK_{k+1}^{(2)}) \\ &= B_{n-k-1}^{(1)} = B_{n-(k+1)}^{(5-4)} \end{aligned}$$

$$DB_{k+1}^{(2)} = DB_k^{(3)} = B_{n-k}^{(2)} = B_{n-k-1}^{(3)} = B_{n-(k+1)}^{(5-2)}$$

$$\begin{aligned} DB_{k+1}^{(3)} &= DB_k^{(2)} + f(DB_k^{(3)}, DK_{k+1}^{(1)}) \\ &= B_{n-k}^{(3)} \oplus f(B_{n-k}^{(2)}, DK_{k+1}^{(1)}) \\ &= B_{n-k-1}^{(2)} \oplus f(B_{n-k-1}^{(3)}, K_k^{(1)}) \oplus f(B_{n-k-1}^{(3)}, DK_{k+1}^{(2)}) \\ &= B_{n-k-1}^{(2)} = B_{n-(k+1)}^{(5-3)} \end{aligned}$$

Therefore, $DB_n^{(j)} = B_{n-j}^{(j)}$, for $1 \le i \le n$ and $1 \le j \le 4$

Q.E.D

Applying Eq.(3) with $i = 1$, we get $DB_n^{(4)} = B_0^{(1)}$, $DB_n^{(4)} = B_0^{(1)}$, $DB_n^{(4)} = B_0^{(1)}$ and $DB_n^{(4)} = B_0^{(1)}$. So $IP(DB_n^{(4)}, DB_n^{(3)}, DB_n^{(2)}, DB_n^{(1)})$ is exactly the plaintext.

## 2.3 Discussion

According to the description above, we know that variant I is still a Feistel network[1] like DES. The operations of each round are in fact two one-round operations of DES. On the last round the resulting four sub-blocks are the reversed order of the previous rounds.

In the encryption process of variant I, the 128-bit plaintext is divided into four initial sub-blocks $B_0^{(1)}, B_0^{(2)}, B_0^{(3)}$ and $B_0^{(4)}$. Then the four sub-blocks are mixed together through repeated rounds to make each of output sub-block after the encryption process is dependent on all the initial sub-blocks. We use a function $S(B_i^{(j)})$ to present the set that influences $B_i^{(j)}$.

**Definition 2.1**

$$S(B_i^{(j)}) = \left\{ B_k^{(l)} \middle| B_i^{(j)} \text{ is in fluenced by } B_k^{(l)} \right\}$$

Hence, the influence relation of each sub-block in each round by the initial sub-blocks are:

Round1:

$$S(B_1^{(1)}) = \left\{B_0^{(2)}\right\}, S(B_1^{(2)}) = \left\{B_0^{(1)}, B_0^{(2)}\right\}$$
$$S(B_1^{(3)}) = \left\{B_0^{(4)}\right\}, S(B_1^{(4)}) = \left\{B_0^{(3)}, B_0^{(4)}\right\}$$

Round 2:

$$S(B_2^{(1)}) = S(B_1^{(4)}) = \left\{B_0^{(3)}, B_0^{(4)}\right\}$$
$$S(B_2^{(2)}) = S(B_1^{(3)}) = \left\{B_0^{(4)}\right\}$$
$$S(B_2^{(3)}) = S(B_1^{(2)}) \cup S(B_1^{(3)}) = \left\{B_0^{(1)}, B_0^{(2)}, B_0^{(4)}\right\}$$
$$S(B_2^{(4)}) = S(B_1^{(1)}) \cup S(B_1^{(4)}) = \left\{B_0^{(2)}, B_0^{(3)}, B_0^{(4)}\right\}$$

Round 3:

$$S(B_3^{(1)}) = S(B_2^{(2)}) = \left\{B_0^{(4)}\right\}$$
$$S(B_3^{(2)}) = S(B_2^{(1)}) \cup S(B_2^{(2)}) = \left\{B_0^{(3)}, B_0^{(4)}\right\}$$
$$S(B_3^{(3)}) = S(B_2^{(4)}) = \left\{B_0^{(2)}, B_0^{(3)}, B_0^{(4)}\right\}$$
$$S(B_3^{(4)}) = S(B_2^{(3)}) \cup S(B_2^{(4)}) = \left\{B_0^{(1)}, B_0^{(2)}, B_0^{(3)}, B_0^{(4)}\right\}$$

Round 4:

$$S(B_4^{(1)}) = S(B_3^{(4)}) = \left\{B_0^{(1)}, B_0^{(2)}, B_0^{(3)}, B_0^{(4)}\right\}$$
$$S(B_4^{(2)}) = S(B_3^{(3)}) = \left\{B_0^{(2)}, B_0^{(3)}, B_0^{(4)}\right\}$$
$$S(B_4^{(3)}) = S(B_3^{(2)}) \cup S(B_3^{(3)}) = \left\{B_0^{(2)}, B_0^{(3)}, B_0^{(4)}\right\}$$
$$S(B_4^{(4)}) = S(B_3^{(1)}) \cup S(B_3^{(4)}) = \left\{B_0^{(1)}, B_0^{(2)}, B_0^{(3)}, B_0^{(4)}\right\}$$

Round 5:

$$S(B_5^{(1)}) = S(B_4^{(2)}) = \left\{B_0^{(2)}, B_0^{(3)}, B_0^{(4)}\right\}$$
$$S(B_5^{(2)}) = S(B_4^{(1)}) \cup S(B_2^{(2)}) = \left\{B_0^{(1)}, B_0^{(2)}, B_0^{(3)}, B_0^{(4)}\right\}$$
$$S(B_5^{(3)}) = S(B_4^{(4)}) = \left\{B_0^{(1)}, B_0^{(2)}, B_0^{(3)}, B_0^{(4)}\right\}$$
$$S(B_5^{(4)}) = S(B_4^{(3)}) \cup S(B_2^{(4)}) = \left\{B_0^{(1)}, B_0^{(2)}, B_0^{(3)}, B_0^{(4)}\right\}$$

Round 6:

$$S(B_6^{(1)}) = S(B_5^{(2)}) = \left\{B_0^{(1)}, B_0^{(2)}, B_0^{(3)}, B_0^{(4)}\right\}$$
$$S(B_6^{(2)}) = S(B_5^{(1)}) \cup S(B_2^{(2)}) = \left\{B_0^{(1)}, B_0^{(2)}, B_0^{(3)}, B_0^{(4)}\right\}$$
$$S(B_6^{(3)}) = S(B_5^{(4)}) = \left\{B_0^{(1)}, B_0^{(2)}, B_0^{(3)}, B_0^{(4)}\right\}$$
$$S(B_6^{(4)}) = S(B_5^{(3)}) \cup S(B_2^{(4)}) = \left\{B_0^{(1)}, B_0^{(2)}, B_0^{(3)}, B_0^{(4)}\right\}$$

The derivation results are listed in table 2.4. From the table we see that after six rounds each of the four output sub-blocks is influenced by all the initial sub-blocks. Therefore, the number of rounds must be greater than or equal to six.

| | Sub-block 1 | Sub-block 2 | Sub-block 3 | Sub-block 4 |
|---|---|---|---|---|
| Round 1 | $B_2^{(2)}$ | $B_2^{(1)}, B_0^{(2)}$ | $B_2^{(4)}$ | $B_0^{(3)}, B^{(4)}$ |
| Round 2 | $B_2^{(1)}, B_2^{(3)}$ | $B_2^{(4)}$ | $B_2^{(1)}, B_2^{(2)}, B_2^{(4)}$ | $B_0^{(2)}, B_0^{(3)}, B^{(4)}$ |
| Round 3 | $B_2^{(2)}$ | $B_2^{(3)}, B_2^{(4)}$ | $B_0^{(2)}, B_2^{(1)}, B_2^{(4)}$ | $B_0^{(1)}, B_0^{(2)}, B^{(4)}, B^{(4)}$ |
| Round 4 | $B_0^{(1)}, B_0^{(2)}, B_0^{(3)}, B_0^{(4)}$ | $B_2^{(2)}, B_2^{(3)}, B_2^{(4)}$ | $B_0^{(2)}, B_2^{(1)}, B_2^{(4)}$ | $B^{(1)}, B_0^{(2)}, B^{(4)}, B^{(4)}$ |
| Round 5 | $B_0^{(2)}, B_0^{(3)}, B_0^{(4)}$ | $B_0^{(1)}, B_0^{(2)}, B_0^{(3)}, B_2^{(4)}$ | $B_0^{(1)}, B_0^{(2)}, B_0^{(3)}, B_0^{(4)}$ | $B_0^{(1)}, B_0^{(2)}, B^{(4)}, B^{(4)}$ |
| Round 6 | $B_0^{(1)}, B_0^{(2)}, B_0^{(3)}, B_0^{(4)}$ | $B_0^{(1)}, B_0^{(2)}, B_0^{(3)}, B_2^{(4)}$ | $B_0^{(1)}, B_0^{(2)}, B_0^{(3)}, B_0^{(4)}$ | $B_0^{(1)}, B_0^{(2)}, B^{(4)}, B^{(4)}$ |

Table 2.4 The relation between sub-blocks in each round and the initial sub-blocks

## 2.4 Extended variant I

The block size of plaintext in variant I can be expanded to $32 \times q$ ($q$ is even and greater than or equal to 4) with one 56-bit key. We call it extended variant I. In this case, the initial and final permutation are ignored and the whole algorithm is as follows:

***Algorithm 2.2*** Encryption of extended variant I

Input: $32 \times q$-bit plaintext $P$, subkeys $K_1, K_2, ..., K_n$

Output: $32 \times q$-bit ciphertext $C$

Process:

$P$ is divided into $B_0^{(1)}, B_0^{(2)}, ..., B_0^{(q)}$

for $i$=1 to $n$

Begin

if $i$ is odd

Begin

for $j$=1,3,5,7,...,$q$-1,

Begin

$B_i^{(j)} = B_{i-1}^{(j+1)}$

$B_i^{(j+1)} = B_{i-1}^{(j)} \oplus f(B_{i-1}^{(j+1)}, K_i)$

End

End

else

Begin

$B_i^{(1)} = B_{i-1}^{(q)}$

$B_i^{(q)} = B_{i-1}^{(1)} \oplus f(B_{i-1}^{(q)}, K_i)$

$j$= 2, 4, 6, 8, ..., $q$-2,

Begin

$B_i^{(j)} = B_{i-1}^{(j-1)}$

$B_i^{(j-1)} = B_{i-1}^{(j)} \oplus f(B_{i-1}^{(j-1)}, K_i)$

End
End
End
$C = (B_n^{(q)}, B_n^{(q-1)}, ..., B_n^{(1)})$

□

Similarly, the number of round must be even and after the last round, the resulting sub-blocks are reserved and combined together to form the ciphertext. This is done to make the encryption and decryption algorithms the same.

In DES, it enciphers 64-bit block of plaintext; our extended variant I enciphers $32 \times q$-bit block for $q$ is unfixed but must be even and greater than 4. That is, the number of sub-block is unfixed and therefore expanded variant I is more flexible. As $q$ increases, the encryption is more secure, but the number of rounds must be increased too.

## 3. Variant II

### 3.1 Encryption

The second proposed variant, we call it variant II in Fig 3.1, operates on a multiple 32-bit blocks with 56-bit key. The 56-bit key is used to generate $n$ ($n$ is the number of rounds) subkeys and the generation algorithm of subkeys is the same as variant I. Initially, the block is divided into $q$ 32-bit sub-blocks. There are $n$ rounds with identical operations. In the first round, $B_1^{(1)}$ is the same as $B_0^{(2)}$. The sub-block $B_0^{(3)}$ is operated with the key, $K_1$, for this round under function $f$ and XORed with sub-block $B_0^{(3)}$ to get the sub-block $B_1^{(2)}$. The sub-block $B_0^{(3)}$ is combined with the key $K_1$ under function $f$ and XORed with sub-block $B_0^{(4)}$ to get the sub-block $B_1^{(3)}$....etc. The sub-block $B_0^{(q)}$ is combined with the key, $K_1$, under function $f$ and XORed with the sub-block $B_0^{(1)}$ to get the sub-block $B_1^{(q)}$. These operations are repeated $n$ times.

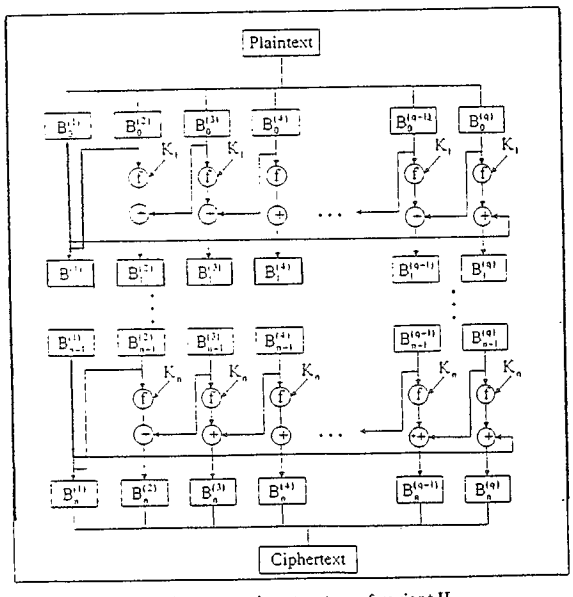


Figure 3.1 The encryption structure of variant II

The encryption algorithm is as follows:

*Algorithm 3.1* Encryption of variant II
Input : Plaintext P, subkey $K_1, K_2, ..., K_n$
Output: Ciphertext C
Process:
$P$ is divided into $B_0^{(1)}, B_0^{(2)}, ..., B_0^{(q)}$
for $i$ = 1 to $n$
Begin
$B_i^{(1)} = B_{i-1}^{(2)}$
for $j$ = 2 to $q$-1
$B_i^{(j)} = B_{i-1}^{(j+1)} \oplus f(B_{i-1}^{(j)}, K_i)$
$B_i^{(q)} = B_{i-1}^{(1)} \oplus f(B_{i-1}^{(q)}, K_i)$
End
$C = (B_n^{(1)}, B_n^{(2)}, ..., B_n^{(q)})$

□

### 3.2 Decryption

The decryption structure is given in Figure 3.2. The ciphertext is first divided into $q$ 32-bit sub-blocks and then passed through $n$ rounds with identical operations. The subkeys used here are the reverse of the encryption subkeys and they are $K_n, K_{n-1}, ..., K_1$. The relation between the encryption and decryption subkeys is :

$$DK_i = K_{n+1-i}, \text{ for } 1 \le i \le n \quad (4)$$

In the round $i$, $1 \le i \le n$, the sub-block $DB_i^{(2)}$ is the same as $DB_{i-1}^{(1)}$. Then $DB_i^{(2)}$ is combined with the key for this round $DK_i$ under function $f$ and XORed with the sub-block $DB_{i-1}^{(2)}$ to get the sub-block $DB_i^{(3)}$. The new generated sub-block $DB_i^{(3)}$ is combined with the key $DK_i$ under function $f$, XORed with the sub-block $DB_{i-1}^{(3)}$ and gets the sub-block $B_i^{(4)}$,...etc. The new $q$th sub-block $DB_i^{(q)}$ is combined with the key for this round under function $f$, XORed with the old $q$th sub-block $DB_{i-1}^{(q)}$ and gets the new first sub-block $DB_i^{(1)}$.

The decryption algorithm for variant II is given as follows:

*Algorithm 3.2* Decryption of variant II

Input : Ciphertext $C$,

subkey $DK_1, DK_2, ..., DK_n$

Output : Plaintext $P$

Process :

$C$ is divided into $DB_0^{(1)}, DB_0^{(2)}, ..., DB_0^{(q)}$

For $i$ = 1 to $n$

Begin

$DB_i^{(2)} = DB_{i-1}^{(1)}$

For $j$=3 to $q$

$DB_i^{(j)} = DB_{i-1}^{(j-1)} \oplus f(DB_i^{(j-1)}, DK_i)$

$DB_i^{(1)} = DB_{i-1}^{(q)} \oplus f(DB_i^{(q)}, DK_i)$

End

$P = (DB_n^{(1)}, DB_n^{(2)}, ..., DB_n^{(q)})$

□

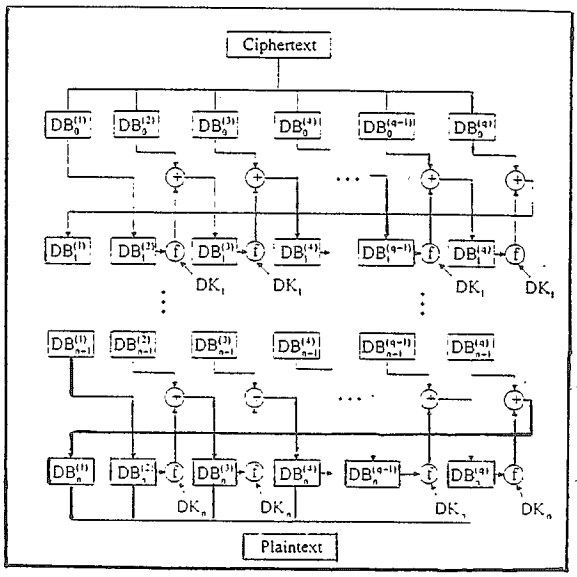


Figure 3.2 The decryption structure of variant II

Note that the decryption operations of one round are sequential, which means that the $q$ blocks of one round can not be processed with parallel. For example, the sub-block $DB_i^{(3)}$ must be generated before the sub-block $DB_i^{(4)}$ ; the sub-block $DB_i^{(4)}$ must be generated before the new fifth sub-block $DB_i^{(5)}$ ,...etc.

*Theorem 3.1* The relations between the encryption and decryption sub-blocks are as follows:

$$DB_0^{(j)} = B_n^{(j)} \text{ for } 1 \le j \le q \quad \text{and} \quad (5)$$

$$DB_i^{(j)} = B_{n-i}^{(j)} (1 \le j \le q, 1 \le i \le n). \quad (6)$$

*proof:*

(1)We will prove it by using mathematical induction.

Basic of induction : $i$ =1, we will prove

$DB_1^{(j)} = B_{n-1}^{(j)}$ for $j$ = 2, 3, ..., $q$.

Clearly when $j$=2, $DB_1^{(2)} = DB_0^{(1)} = B_n^{(1)} = B_{n-1}^{(2)}$.

Suppose $DB_1^{(k)} = B_{n-1}^{(k)}$.

We will prove $DB_1^{(k+1)} = B_{n-1}^{(k+1)}$

$DB_1^{(k+1)} = DB_0^{(k)} \oplus f(DB_1^{(k)}, DK_1)$

$= B_{n-1}^{(k+1)} \oplus f(B_{n-1}^{(k)}, K_n) \oplus f(B_{n-1}^{(k)}, DK_1) = B_{n-1}^{(k+1)}$

Therefore, $DB_1^{(j)} = B_{n-1}^{(j)}$ for $2 \le j \le q$.

For $j$=1,

$DB_1^{(1)} = DB_0^{(q)} \oplus f(DB_1^{(q)}, DK_1)$

$= B_{n-1}^{(1)} \oplus f(B_{n-1}^{(q)}, K_n) \oplus f(B_{n-1}^{(q)}, DK_1) = B_{n-1}^{(1)}$

(2) Suppose it is true for $i = r$ , i.e.,

$DB_r^{(j)} = B_{n-r}^{(j)} (1 \le j \le q)$.

(3) Consider $i = r+1$,

When $j$=2, $DB_{r+1}^{(2)} = DB_r^{(1)} = B_{n-r}^{(1)} = B_{n-r-1}^{(2)}$

Suppose that it is true for $j = k$, i.e., $DB_r^{(k)} = B_{n-r}^{(k)}$.

When $j = k+1$,

$DB_{r+1}^{(k+1)} = DB_r^{(k)} \oplus f(DB_{r+1}^{(k)}, DK_{r+1})$

$= B_{n-r-1}^{(k+1)} \oplus f(B_{n-r-1}^{(k)}, K_{n-1}) \oplus f(B_{n-r-1}^{(k)}, DK_{r+1})$

$= B_{n-r-1}^{(k+1)}$

Therefore, $DB_{r+1}^{(j)} = B_{r+1}^{(j)}$ for $2 \le j \le q$

For $j$ = 1,

$DB_{r+1}^{(1)} = DB_r^{(q)} \oplus f(DB_{r+1}^{(q)}, DK_{r+1})$

$= B_{n-r-1}^{(1)} \oplus f(B_{n-r-1}^{(q)}, K_{n-1}) \oplus f(B_{n-r-1}^{(q)}, DK_{r+1})$

$= B_{n-2}^{(1)}$

From (1), (2), (3), we have the results.

Q.E.D

Applying Eq.(6) with $i = n$, we get

Ciphertext = $(DB_n^{(1)}, DB_n^{(2)}, ..., DB_n^{(q)}) = (B_0^{(1)}, B_0^{(2)}, ..., B_0^{(q)})$

i.e., the decryption result is the plaintext.

## 3.3 Discussion

Variant II operates on $32 \times q$-bits blocks which make the cipher more secure. The output bit is influenced by $q$ blocks rather than two blocks in DES.

The number of sub-block, $q$, depends on the size of the plaintext. It equals the size of plaintext divided by 32. In the algorithm, the round numbers are influenced by block number $q$.

The number of round, $n$, is a variable which is influenced by $q$ and affects the security of the cipher. As $q$ increases, $n$ should also increase to make the cipher secure. The design of the key scheduling is similar to the one of DES except the number of left circular shift per round must change. We have already discussed the concept in section 2.1.

Comparing the encryption algorithm of variant II with GDES we mentioned before,

every sub-block, except $B_i^{(2)}$, in variant II is operated with subkey and then XORed with the sub-block right next to it. In GDES, only one sub-block is operated with subkey. So, variant II is more secure.

### 3.4 Observation

In observing the original DES and our variants, we find out that there is always some sub-block remaining the same in each round and we use this sub-block to make decryption process work properly. If we design a variant of DES that does not keep this property, which means that none of the sub-blocks in each round remain the same, there does not exist any decryption algorithms.

## 4. Conclusion

In this paper, we discuss several cross-block DES. The original DES process 64-bit block of plaintext. Here, we process blocks of variable size. We design two variants. All of the variants have variable round number. The variant I processes 128-bit block of plaintext with 112-bit key and can expand to process $64 \times q$-bit of plaintext. Every two sub-blocks are operated together in one round. The variant II processes $32 \times q$-bit block of plaintext with 56-bit key. The second input sub-block keeps the same in the round; all the remaining sub-blocks are operated with the subkeys and XORed with the next right sub-blocks. We also find out that there does not exist decryption algorithms for some variants of the cross-block.

## Reference

[1] Bruce Schneier, "*Applied Cryptography*", John Wiley & Sons, Inc., 1996.

[2] Schaumuller-Bichl, "*On the Design and Analysis of New Cipher Systems Related to the DES*," technical Report, Linz University, 1983.